

DNELSON
66

No. 19

Revised June 1965

USER'S GUIDE

MAN-MACHINE INFORMATION SYSTEM



STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA

STANFORD RESEARCH INSTITUTE

MENLO PARK, CALIFORNIA



Revised June 1965

USER'S GUIDE

MAN-MACHINE INFORMATION SYSTEM

The Man-Computer Program at Stanford Research Institute is one of the activities within the Systems Engineering Laboratory of the Engineering Sciences and Industrial Development Division. The personnel participating in the program are:

Supervisory

J. D. Noe, Executive Director
Engineering Sciences and Industrial Development

Torben Meisling, Manager
Systems Engineering Laboratory

Full Time Participants in Program

D. C. Engelbart (Head of Program)

M. L. Berman
W. K. English
Bonnie Huddart
T. L. Humphrey
N. T. Ward
J. M. Yarborough

Full Time Clerical, Secretarial and Operator Support

Patricia Conley
Judy Sass

Steady Part Time Participants in Program

R. D. Bates
G. C. Chaplin
C. R. Kirkley
A. R. Tobey

I	INTRODUCTION	A-5
II	LINKED-STATEMENT STRUCTURING: TERMINOLOGY AND CONVENTIONS	A-7
III	ON-LINE TEXT MANIPULATION SYSTEM	A-17
	A. Introduction	A-17
	B. Operating Procedures	A-19
	C. On-Line Computer Equipment	A-27
IV	OFF-LINE TEXT MANIPULATION SYSTEM	A-35
	A. Introduction	A-35
	B. Statement-Structure Modification	A-37
	1. Procedures for Entering Material	A-37
	2. Preparing Copy	A-40
	3. User Guide	A-47
	4. Operator Instructions	A-53
	C. Editing Within Statements: The Z-Code System	A-56
	1. Z-Code Reference Summary	A-56
	2. Teletype Conventions	A-63

1 The Man-Machine Information System is aimed at improving the work performance of a programmer by the use of computer aids, many of them real-time. Although a programmer is the main target for the work, many of the processes have a wider applicability.

2 This manual describes the current state of the system, which is in continuous development. The manual is published in two forms--in a looseleaf notebook and in a fixed binding.

2a The looseleaf form is intended for those who will use the system.

2a1 Such users will receive, when appropriate, modifications and additions to this manual to keep their copy updated with the status of the system.

2b The fixed binding version of this manual is for information purposes only. New versions will be issued from time to time as significant additions are incorporated into the system.

2c Requests for further copies of this manual in either version should be made to Mr. W. K. English, Building 314b.

3 The stimulus for the design of the system has been the Institute's research program on "Augmented Human Intellect."

3a The initial conceptual framework for the Augmented Human Intellect Study was supported jointly by the Air Force Office of Scientific Research (AF 49(638)-1024) and Stanford Research Institute over the period from 1961 onwards.

4 In this manual, the system is broken down into several components that have been developed under various contracts to form a coordinated whole.

4a This particular version of the User's Guide is assembled specifically to accompany the March 19, 1965 report of the ESD project.

4b The sections describing the conventions and procedures

SECTION I -- INTRODUCTION

for program-design documentation are missing in favor of the more complete writeup in the report itself.

5 The contents are arranged in the following categories:

5a The conventions, concepts, and definitions for the linked-statement structure form.

5a1 Essentially all of our text is now composed and manipulated in this form; thus our computer-aided processes are oriented specifically toward manipulating this form of text (although they will also handle other forms).

5b The procedures and processes available on our on-line system for manipulating our working text, and the equipment comprising our on-line facility.

5c The procedures and processes available on our off-line system for manipulating our working text.

6 Currently both the off-line and on-line systems work with paper-tape input and output.

6a The paper-tape output is the result of the operations done upon the paper-tape input text and can be printed on the Flexowriter to obtain corresponding hard copy.

6b The paper-tape output of either the on-line or the off-line system is compatible as input to either system for a next stage of manipulation.

SECTION II -- LINKED-STATEMENT STRUCTURING:
TERMINOLOGY AND CONVENTIONS

1 These conventions and terminology for linked-statement structuring were developed under the sponsorship of the Advanced Research Projects Agency.

2 Statements.

2a Any appearance of the sequence CARRETURN CARRETURN NUMERIC is assumed to signal the beginning of a new statement, with the NUMERIC as the first character of the first "word."

2b The length of a statement is arbitrary.

2c The composition of a statement is arbitrary, with the following explicit exceptions:

2c1 Special requirements for Location Numbers see(LOCNUMDF), Names see(NAMDEF), Tags see(TAGDEF), and Links see(LINKDEF) are described below.

2d (LOCNUMDF) Location numbers.

2d1 The first word of a statement is its location number; its first character is a digit.

2d2 The location number is composed of a string of digits and alphabets, with no spacing gaps included.

2d2a A "field" in the location number is a continuous string of alphabetic characters, or a continuous string of numeric characters, broken possibly by a period or a comma.

2d2b The characters in a given field indicate the ordering on a unique list in the structure of statements see(STRUCDEF).

2d3 The location number represents the unique location of its statement within the structure of statements.

2e (NAMDEF) Names.

2e1 A name may be associated with any given statement.

2e2 The name is enclosed in parentheses and is the first printing string after the location number.

2e3 If an open paren is the first printing character

SECTION II -- LINKED-STATEMENT STRUCTURING:
TERMINOLOGY AND CONVENTIONS

after the location number, it is assumed to signal the presence of a name.

2e4 The name may contain no spacing gaps--i.e., there will be no spacing gaps between the parentheses.

2e5 The choice and sequence of printing characters composing a name is arbitrary.

2e6 The length of a name is limited to 16 characters (printing or non-printing). This is an arbitrary and tentative limit.

2f (TAGDEF) Special words called "tags" may be included within a statement; they may serve as descriptors, etc.

2f1 As many tags as desired may be included within a statement.

2f2 They may be located anywhere after the location number and name.

2f3 Each is identified by the sequence SPACINGAP
ASTERISK n-PRINTCHARS SPACINGAP.

2f4 There is no restriction on "n," or on the composition of a tag--except that no spacing gaps may be included.

2g (LINKDEF) Special words called "links" may be included within a statement; they serve to establish cross-reference linkages to other statements.

2g1 As many links as desired may be included within a statement.

2g2 They may be located anywhere after the location number and name.

2g3 Each is identified by the sequence SPACINGAP
n-PRINTCHARS OPENPAREN m-PRINTCHARS CLOSEPAREN
SPACINGAP-OR-PUNCTUATION.

2g4 The parens enclose the name of some statement, see(NAMDEF).

2g5 The PRINTCHARS preceding the OPENPAREN represent the "link type" code string; this string may be of arbitrary length and composition-- except that no spacing gaps may be included.

3 (LISTDEF) Lists of Statements.

3a Any statement ST_i may have a "list successor," which is another statement.

3b The sequential string of statements formed by the successor of a statement, by its successor, etc., until finally a statement is reached that has no list successor, is called a "list of statements."

3c The first statement on such a sequential list of statements is called the "head statement" of the list.

3d The last statement on such a sequential list of statements is called the "tail statement" of the list.

3e A list may contain an arbitrary number of statements, but must have at least one statement.

3f For each statement in a given list, the last field of the location number indicates the statement's location in that list.

3fl Interpolative breaks (e.g., 2fl.5) may appear in a field of the location number; in this case the numbers indicate only the relative location.

3fla A special interpolation convention is needed in order to insert something before the head statement of a list.

3flb Let a COMMA, when used as an interpolative break in a field, designate that the interpolation is to come before (rather than after) the statement indicated by the field characters up to the interpolative break.

3flc Example: 3bl,5 (or: 2a,e) would belong in front of, and at the same list level as, 3bl (or: 2a).

3f2 A list in which the location numbers are in "clear ordinal" state will have no interpolative breaks in the last field; this field will then indicate the true ordinal location in the list.

4 (STRUCDEF) List Structures of Statements.

4a Various structural relations are already implied:

SECTION II -- LINKED-STATEMENT STRUCTURING:
TERMINOLOGY AND CONVENTIONS

- 4a1 Sequential association within a list.
- 4a2 Inter-statement links, see(LINKDEF).
 - 4a2a Any statement may be linked to any other in this manner.
- 4b Besides this, there is hierarchical structuring.
 - 4b1 Each list of statements may be a sublist of one (and only one) statement.
 - 4b2 That statement is known as the "source statement" of that list.
 - 4b3 The location number of every statement on such a list will differ from that of its source statement only by the addition of one more field.
 - 4b4 Any statement in that list may be the source statement for a sublist of its own, etc., to arbitrary depth.
 - 4b5 The sublist of a statement, and the sublists of the sublist statements, etc., form the "substructure" of the given statement.
- 5 Terminology Conventions.
 - 5a About the choice of mnemonics: each entity described below has a name that is generally accepted and usually easy to remember. The three-character mnemonic term for designating an entity is derived from this name by means of the following rules:
 - 5a1 The case of any alphabetic character within a mnemonic is not significant.
 - 5a2 For a one-word name, take the first three non-repeated, non-silent consonants.
 - 5a3 If there are not enough consonants, include the first phonetic vowels, ordered with the consonants as they appear in the word.
 - 5a4 For a two-word name, take two characters from the first word, and a third character from the second word, according to the two rules above.
 - 5a5 For a word and a number, take two characters of the

word (as above) and append the number--even if the number is several characters.

5a6 For two words and a number, take one character from each word, and append the number--even if the number is several characters.

5a7 If two names would produce the same 3-character mnemonic, use this mnemonic for the name which precedes alphabetically. For the other mnemonic, try rejecting its second character and picking another character, for a new second or third character, according to the selection rules above.

5b Basic Entities.

5b1 Let ST1, ST2, etc., refer to arbitrary statements.

5b1a The integers carry no implications as to the structural relationship between the statements.

5b2 Let LN1, LN2, etc., be used to represent arbitrary location numbers.

5b3 Let 1F1, 1F2, etc., refer to the first, second, etc., fields of LN1; and 2F1, 2F2, etc., to the first, second, etc., fields of LN2.

5b4 Let NM1, NM2, etc., refer to arbitrary statement names.

5b5 Let LS1, LS2, etc., represent arbitrary lists of statements.

5c Operations--where an operation on one entity represents another entity.

5c1 General:

5c1a Let LCN ST1, LCN ST2, etc., represent the location numbers of statements ST1, ST2, etc.

5c1b Let STM LN1, STM LN2, etc., represent the statements whose location numbers are LN1, LN2, etc.

5c1c Let STM NM1, STM NM2, etc., represent the statements whose names are NM1, NM2, etc.

5c1d Let NAM ST1, NAM ST2, etc., represent the names of statements ST1, ST2, etc.

SECTION II -- LINKED-STATEMENT STRUCTURING:
TERMINOLOGY AND CONVENTIONS

5c1d1 Let NAM ST1 be ZERO if ST1 has no name.

5c2 Fields within a location number:

5c2a Let FL1 LN1, FL2 LN1, etc., represent the first, second, etc., fields of location number LN1.

5c2b Let FL(expression) LN1 represent the nth field of LN1, where n is the numeric obtained by evaluating the expression.

5c2c Let FLi LN1, FLj LN1, etc., refer to the ith, jth, etc., fields of LN1.

5c2d Let FLT LN1 represent the last (tail) field of LN1.

5c3 The depth of a statement--the level down from the top of the structure at which it lies--is an integer. The topmost level (location numbers of 1,2,etc.) has a depth of 1; the next level down (location numbers of 1b, 4d, etc.) has a depth of 2, etc.

5c3a Let DPT ST1, DPT ST2, etc., represent the depths of ST1, ST2, etc.

5c3b Let DPT LN1, DPT LN2, etc., represent the depths of STM LN1, STM LN2, etc.; these should always be equal to the number of fields in LN1, LN2, etc.

5c4 To represent a statement having a particular structural relationship to another statement:

5c4a SCS ST1, successor of ST1 (list successor).

5c4b PRD ST1, predecessor of ST1 (list predecessor).

5c4c HED ST1, head of the list containing ST1.

5c4d TAL ST1, tail of the list containing ST1.

5c4e SBH ST1, sublist head of ST1--the head statement of the sublist of ST1.

5c4f SBT ST1, sublist tail of ST1--the tail statement of the sublist of ST1.

5c4g SRC ST1, source of ST1--the source statement of

ST1.

5c5 To represent a list having a particular structural relationship to a statement:

5c5a LSC ST1, list containing ST1--the entire list of statements.

5c5b LSF ST1, list from ST1--the list of statements including ST1, SCS ST1, etc., down to and including TAL ST1.

5c5c LSB ST1 ST2, list between ST1 and ST2--a binary operation, representing the list that begins with ST1 and ends with ST2. (ST1 and ST2 must be in the same list.)

5c5d LST ST1, list to ST1--the list of statements from HED ST1 through PRD ST1.

5c5e SBL ST1, sublist of ST1--the entire list.

5c5f SRL ST1, source list of ST1--the list containing SRC ST1.

5c6 To represent a statement having a particular relationship to a list:

5c6a HED LSl, head of LSl.

5c6b TAL LSl, tail of LSl.

5c6c SRC LSl, source of LSl.

5c7 Relating a list to a list:

5c7a SRL LSl, source list of LSl--the list containing SRC LSl.

5d Concatenated operations.

5d1 Notation:

5d1a An operator may operate upon an entity that is represented as the product of another operation.

5d1b Two successive operator terms separated by a spacing indicate that the entity represented by the rightmost operation is to be operated upon by the preceding operator term.

SECTION II -- LINKED-STATEMENT STRUCTURING: TERMINOLOGY AND CONVENTIONS

5dlc Obviously, the product of the rightmost operation must be an entity upon which the preceding operator can validly operate.

5dl d An integer n , or an expression representing such an integer, appearing between parentheses after an operator, designates n successive applications of that operator.

5dle Any other printing character or characters appearing between two operations indicates that they are not to be concatenated.

5dlf Some reasons for this notation:

5dlf1 Spacingaps between concatenated terms are desirable so that long chains can be conveniently broken by line spacing without any complications.

5dlf2 Prefix Polish notation offers a good precedent. (So does suffix notation--we arbitrarily selected prefix.)

5d2 Examples:

5d2a LCN TAL SRC ST1 is the location number of the tail statement of the list containing the source statement of ST1.

5d2b SCL ST1 = LSC SRC ST1.

5d2c SBL ST1 = LSF SBH ST1.

5d2d LST ST1 = LSB HED ST1 PRD ST2.

5d2e FL(DPH LCN ST2) LCN ST1 is the field of LCN ST1 at a depth corresponding to the last field of LCN ST2.

5e Special entities and relationships:

5e1 The "source chain" of ST1 is composed of ST1, SRC ST1, SRC(2) ST1, ..., SRC(DPT ST1) ST1.

5e2 The "branch chain" from ST1 is composed of LST ST1, tied onto the end of LST SRC ST1, tied onto the end of LST SRC(2) ST1, etc., to the head of the top-level list of the structure.

SECTION II -- LINKED-STATEMENT STRUCTURING:
TERMINOLOGY AND CONVENTIONS

5e3 ST1 is said to be "structurally above" ST2 if ST1 is a member of the branch chain from ST2.

5e4 ST1 is said to be "structurally below" ST2 if ST2 is a member of the branch chain of ST1.

5e5 ST1 is said to be "branch related" to ST2 if either statement is a member of the other's branch chain.

5e6 ST1 is said to be "branch independent" of ST1 if neither statement is a member of the other's branch chain (i.e., if they are not branch related).

5e7 ST1 is said to be the "branch node" between statements ST2 and ST3 if it lies in the branch chains of both ST2 and ST3, and if it is below every other statement that does so.

5e7a The branch chains from any two statements in the same structure will always meet to produce such a node.

5e7b The branch node between two branch-related statements will be the "upper" of the two statements--i.e., the one which is structurally above the other.

5e7c Let BRN ST2 ST3 be a symmetrical, binary (two-parameter) operator whose result represents the branch-node statement (e.g., ST1 = BRN ST2 ST3 = BRN ST3 ST2).

5e8 The "bridge chain" from ST1 and ST2 is the concatenation of the section of the branch chain of ST1 from ST1 to BRN ST1 ST2, with the section of branch chain of ST2 from BRN ST1 ST2 to ST2.

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART A -- INTRODUCTION

1 Various segments of this on-line system have been developed under different sponsorship, according to the pursuits of the respective projects.

1a The basic working system was developed and programmed under the sponsorship of the Advanced Research Projects Agency. This includes the routines for storing data on drum and tape; for inputting and outputting; and for executing the higher-level commands that operate on statement structures and tape files.

1b A project from the National Aeronautics and Space Administration developed and programmed those parts of the basic operating system that handle the core-held "current data"; the interface and interpretive routines that service the display and command-designation operations; and the basic editing routines.

2 With this system, one can load an arbitrary number of working records (each up to 18,000 characters in length) onto magnetic tape by typing at the on-line keyboard, or by reading in paper tape from any of our paper-tape-punching typewriters or from the output of our off-line system.

2a The system will handle a variety of text forms (including the normal sentence-paragraph form), but a number of its special features are specifically designed for the linked-statement form.

3 With the CRT display as a very mobile "window" to scan a record, and with the computer to maneuver the window and alter the record in instantaneous response to his directions, the user can study and/or modify any such record with great facility.

4 He may access any of his working records, for study and modification; or make an internal copy, for independent storage and alteration as a new record; or extract from a number of such records, merging them to form a new record.

5 At any time he may punch a record onto paper tape, to be kept permanently if desired. At any later time he may then use this tape to re-enter this information back into the on-line system; to type a printed version on the Flexowriter; or as an input to the off-line system.

6 Once the equipment has been turned on, and the on-line program has been loaded and initiated at the computer, the user

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART A -- INTRODUCTION

directs all further system actions from the work station (the CRT display, keyboard, etc.) by means of successively designated commands.

6a Each command is executed immediately.

6b The function of the commands, individually and collectively, has been designed to be maximally useful in the task environment of working with the linked-statement structures that represent our working records of plans, specifications, computer-program design records, system-reference documents, external-document reference files, report drafts, etc.

6c Each command is designated by a simple, convenient combination of keyboard-character strokes and screen-selection actions (with light pen or table cursor).

6d *c A sizeable portion of our research effort continually goes toward improving the repertoire and designation means of these commands.

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART B -- OPERATING PROCEDURES

1 Various segments of the on-line system have been developed under different sponsorship, according to the pursuits of the respective projects.

1a The basic working system was developed and programmed under the sponsorship of the Advanced Research Projects Agency. This includes the routines for storing data on drum and tape; for inputting and outputting; and for executing the higher-level commands that operate on statement structures and tape files.

1b A project from the National Aeronautics and Space Administration developed and programmed those parts of the basic operating system that handle the core-held "current data"; the interface and interpretive routines that service the display and command-designation operations; and the basic editing routines.

2 The two basic components of a command--the operator and the operands (or parameters).

2a The operator--specifying which command of the repertoire is to be executed.

2a1 Generally designated by several mnemonic alphabetic characters (with perhaps a SPACE stroke) struck by the user on the keyboard. Case of alphabets is unimportant.

2a2 Or, a special one-handed keyboard may be used, leaving the other hand free for light-pen or cursor use. This has specially arranged keys for designating forward or backward scan, and for delete, insert, replace, move, and copy operations on text, character, word, line, and statement entities.

2a3 Full name for operator appears on top line of display immediately after the operation is thus designated.

2a4 After command execution, operation name remains displayed; successive executions do not require re-designating the operation.

2a5 Generally, input characters will be interpreted as command-operation designation only after: a command has just been executed (by striking the CA key), a command has just been aborted (by striking the CD key), or the system has just been started up.

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART B -- OPERATING PROCEDURES

2b The operands and parameters--three types:

2b1 A numerical parameter, e.g., for designating how many lines to scan or which type-file item to access. Entered at appropriate time (see below) from the alphanumeric keyboard.

2b2 Operand entities displayed on the screen.

2b2a Selected by locating the light pen or cursor near a character or printing space and hitting the associated SELECT button.

2b2b User actually selects a character (which can be a non-print character); if a larger entity (i.e., word, line, or statement) is called for as an operand, the computer takes that entity which includes the selected character.

2b3 Literal input, a string of characters entered at the appropriate time on the alphanumeric keyboard.

2b3a Always terminate LIT with a CA.

2b3b At the time during a command designation that LIT is expected by the computer, a space is cleared on the display and the user sees the character-by-character accrual of his keyboard input--to be put in the specified text location by the final CA action.

2b3c During LIT input, a BACKSPACE deletes the last character of the LIT string.

2b3d Similarly, a BACKSPACEWORD (a special key) deletes the last word.

2b3e The user need not be concerned with new-line designation; if a word is being entered and the end of the line is reached before a SPACE is entered, the computer automatically shifts the partial word to the start of the next line.

3 Executing or aborting a command.

3a After designating appropriately the operation, parameters, and operands, striking the CA key (there is one on each side of the keyboard) will cause the command to be executed.

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART B -- OPERATING PROCEDURES

3a1 On the commands not involving a literal input, a SPACE key (generally easier to strike) may be used optionally in place of the CA key.

3a2 A bug-select actuator on a cursor has exactly the same effect as the CA key, and may be used in its stead at any time.

3b At any point in designating a command, striking the CD (command-delete) key will abort the command.

3b1 The operator designation in the top line of the display will remain as it was before hitting the CD key.

4 Many commands change the contents of statement; the new formatting is automatically done by the computer.

4a In general all the text of a statement is cut into new line assignments. A given line is terminated (by a new-line start) at the inter-word gap which comes nearest to filling out a stipulated length of the line.

4b The exception: if a line contains a TAB in it, then its line-start text position remains fixed.

4c On type-out or punch-out, leading SPACE and TAB codes are inserted to indent each line of a statement 3d spaces, where "d" is an integer one less than the structural depth of that statement.

5 Command-description conventions.

5a A description of the way a given command is designated is presented below as a succession of (upper-case) character groups, each separated by a SPACE.

5b The single letters each represent the corresponding single alphabetic character to be entered. (Case is unimportant in actual usage.)

5c SP represents a SPACE character.

5d C1,C2,..., W1,W2,..., L1,L2,..., S1,S2,..., represent user-designated characters, words, lines, or statements--each specified at command-designation time by selection of any single character within the entity.

5e LIT represents a literal-input string and includes all characters entered, even SPACE, TAB, and CARRETURN.

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART B -- OPERATING PROCEDURES

5f NUMBER represents any decimal integer entered on the alphanumeric keyboard.

5g CA represents hitting the CA (command-accept) key.

6 Commands currently available:

6a Summary list:

6a1 Enter text from designated source into working space on drum.

E P CA	Enter from paper tape
E M CA	Enter from currently positioned file on mag tape
E K CA LIT CA	Enter from keyboard--automatically positions display at end of drum's working text, and adds keyboard entry (LIT) character by character to the end

6a1a This new data is added to the end of the existing working data on the drum.

6a1b The "enter" process will halt when drum is near full, and the typewriter will print appropriate notice. This allows for some free space for copying and inserting. Reinitiating the "enter" command will load until working space is full.

6a1c When entering from a mag-tape file, the tape will remain positioned where the "enter" process stopped, and unless disturbed by an intervening tape-file command, a subsequent E M command will continue reading in that file from that point.

6a2 Position display frame on working text of drum.

H N CA LIT CA	Hop to put statement named LIT at top of screen
H P CA LIT CA	Hop to put statement numbered LIT at top of screen
H L W1 CA	W1 a link word, i.e., of form TT..T(LL..L); hop to put statement named LL..L at top of screen.
F S S1 CA	Move forward so as to position statement S1 at top of screen
F S NUMBER SP	Move forward NUMBER statements
F L L1 CA	Move forward so as to position line L1

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART B -- OPERATING PROCEDURES

	at top of screen
F L NUMBER SP	Move forward NUMBER lines
F A CA	Move forward all the way to end of text
B S S1 CA	Move backward so as to position statement S1 at bottom of screen.
B S NUMBER SP	Move backward NUMBER statements
B L L1 CA	Move backward so as to position line L1 three lines from bottom of screen
B L NUMBER SP	Move backward NUMBER lines
B A CA	Move backward all the way to the beginning of text
F B S1 CA	Move forward to next logical break in numbering sequence starting from indicated statement
B B S1 CA	Move backward to next logical break in statement- numbering sequence starting from indicated statement

6a2a See 6a4a for definition of "logical break."

6a3 Modify text seen in display frame.

6a3a Delete the designated entity, and close up the remaining text.

D T C1 C2 CA	Delete text, characters C1 through C2
D C C1 CA	Delete character C1
D W W1 CA	Delete word W1
D L L1 CA	Delete line L1
D S S1 CA	Delete statement S1

6a3b Insert LIT as indicated behind the designated entity. Rearrange prior text as required to make room.

I T C1 LIT CA	Insert LIT after character C1
I C C1 LIT CA	Insert LIT after character C1
I W W1 LIT CA	Insert SPACE LIT after last printing character of word W1
I L L1 LIT CA	Insert CARRETURN LIT after last printing character of line L1
I S S1 LIT CA	Insert CARRETURN CARRETURN LIT after last printing character of statement S1

6a3c Replace the designated entity with LIT, rearranging prior text as necessary.

R T C1 C2 LIT CA	Replace text string characters C1 through C2, with LIT
R C C1 LIT CA	Replace character C1 with LIT

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART B -- OPERATING PROCEDURES

R W W1 LIT CA	Replace word W1 with LIT
R L L1 LIT CA	Replace line L1 with LIT
R S S1 LIT CA	Replace statement S1 with LIT

6a3d Move one designated entity to follow another. The moved entity is deleted from its original location. Other text is adjusted to close the deletion gap and open the corresponding insertion gap.

M T C1 C2 C3 CA	Move the text string, character C2 through C3 to follow character C1
M C C1 C2 C3 CA	Move the text string, character C2 through C3, to follow character C1
M W W1 W2 CA	Move word W2 to follow word W1
M L L1 L2 CA	Move line L2 to follow line L1
M S S1 S2 CA	Move statement S2 to follow statement S1

6a3e Copy one designated entity and insert it behind another. The copied entity remains unchanged. Prior text is rearranged to make room for new insertion.

C T C1 C2 C3 CA	Copy text string, characters C2 through C3, to follow character C1
C C C1 C2 C3 CA	Copy text string, characters C2 through C3, to follow character C1
C W W1 W2 CA	Copy word W2 to follow word W1
C L L1 L2 CA	Copy line L2 to follow line L1
C S S1 S2 CA	Copy statement S2 to follow statement S1

6a4 Renumber successive statements in the working text.

N S1 LIT CA	Give statement S1 the new number LIT, and give successive statements correspondingly appropriate new numbers until a statement ST2 is reached such that either ST2 is of a higher level than S1, or ST2 is not a "logical successor" to the statement preceding it. Display view ends up with the predecessor of ST2 at the top of the frame.
-------------	---

6a4a ST2 is said to be the logical successor to ST3 if there could exist an actual hierarchical structure such that (by their location numbers) ST2 could succeed ST3 in the text. For instance, following 2b3 one could logically accept only 2b3a, 2b4, 2c or 3.

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART B -- OPERATING PROCEDURES

Presence of any other number on the next statement establishes a "logical break" at this point in the text.

6a5 Move or copy statements selected from the display and insert them just before a specified statement somewhere else in the drum-held working text. These operations require a three-character designation.

T S N S1 LIT CA	Transmit (move) S1 to the statement named LIT
T S P S1 LIT CA	Transmit S1 to the place (statement numbered) LIT
T L N S1 S2 LIT CA	Transmit the list of statements S1 through S2 to the statement named LIT
T L P S1 S2 LIT CA	Transmit the list of statements S1 through S2 to the place (statement numbered) LIT
S S N S1 LIT CA	Copy S1 to statement named LIT
S S P S1 LIT CA	Copy S1 to place numbered LIT
S L N S1 S2 LIT CA	Copy list, S1 to S2, to statement named LIT
S L P S1 S2 LIT CA	Copy list, S1 to S2, to place numbered LIT

6a6 Output part or all of the working text to the designated device. The working text remains undisturbed. Three characters are required for operation designation.

O P A CA	Output to punch all working text
O T A CA	Output to typewriter all working text (not yet implemented)
O M A CA	Output to currently positioned mag-tape file all working text, replacing prior contents of that file
O P S S1 S2 CA	Output to punch statements S1 through S2 (S1 may equal S2 for one-statement output)
O T S S1 S2 CA	Output to typewriter, statement
O P P C1 C2 CA	Output to punch partial, characters C1 through C2
O T P C1 C2 CA	Output to typewriter partial, characters C1 through C2

6a7 Clear the working space on the drum of its present contents.

Z W S	Zero work space
-------	-----------------

6a8 Locate and examine tape-file items. Each

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART B -- OPERATING PROCEDURES

fixed-length item space can hold a full drum load of working text, and the items are referenced by decimal-integer serial number corresponding to their order on the tape. Any "look" operation displays the first frameful of text from the tape without either disturbing the drum data or losing the position on tape.

L H CA	Look here, i.e., at text just beyond current position on tape
L I NUMBER CA	Look at item numbered NUMBER--positions tape at head of the item and provides a look
L N CA	Look at next item--the one just beyond the current position
L P CA	Look at prior item--the one just ahead of the current position

6a8a Trying to look beyond the last item, either with L I NUMBER for too large a NUMBER, or with a L N from the very last item of the file, will produce the displayed message, "Beyond last item."

6a8b An O M command at this point will create a new item on the end of the file

6a9 Type out system-status data.

O S CA	Output system status, causes typing in the form: x channels left, item y last read in, tape positioned to item z. "Channels" refer to the 512-character modules of drum working space, of which there are a total of 36.
--------	--

NL EQUIPMENT

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART C -- ON-LINE COMPUTER EQUIPMENT

1 This section contains brief descriptions of the computer and associated peripheral equipment currently used by our on-line text manipulation system.

2 THE COMPUTER (CDC 160A)

2a Memory:

2a1 6.5 usec cycle time.

2a2 12-bit word.

2a3 4,096 words per bank, directly addressable.

2a4 Two banks on our machine--programmer must set up bank controls to shunt his access requests, independently for four categories of access, to the appropriate bank.

2a5 Each bank has independent access circuitry.

2b Instruction repertoire:

2b1 No built-in multiply, divide, square root, etc.

2b2 Full complement of add, subtract, conditional branch, transfer, logic (logical product, selective complement), shifting, input-output, and selective stop and jump (responding to switches on console).

2b3 Since 12 bits can just exactly address 4096 words, all instructions requiring operand specification over a complete bank require two successive words--one for operation specification and one for operand specification.

2b4 A significant proportion of instructions require but one word, and operate with 6 bits of operand specification in one of the following modes:

2b4a Relative forward--addressing one of the 64 words following the cell in which the single-word instruction was located.

2b4b Relative backward--addressing one of the 64 words preceding the cell in which the single-word instruction was located.

2b4c Direct--addressing one of the first 64 words in

a bank specified by the direct-bank bank-control setting.

2b4d Indirect--telling the computer (with a one-word instruction) to go to the specified one of 64 direct-bank words, take the 12-bit contents as the full-bank address of the operand, and look for the operand in the bank specified by the indirect-bank bank control.

2b4e No address--a 6-bit operand is to be found in the lower six bits of the instruction word.

2b5 Variations in the operation code of nearly all the commands indicate which way the operand is to be obtained for that instruction. For example, the add instruction will have the following variations:

2b5a Add no address (adn), add the lower six bits of the instruction word to the accumulator.

2b5b Add direct (add), add to the accumulator the contents of the direct-bank cell specified by the lower six bits of the instruction word.

2b5c Add memory (adm), add to the contents of the accumulator the contents of the memory-bank cell specified by the 12 bits of the word following the instruction word (then get the next instruction from the word following that one). Which bank to use for operand accessing is specified by the setting of the memory-bank control.

2b5d Add indirect (adi), add to the contents of the accumulator the contents of cell in indirect bank that is specified by the contents of the cell in direct bank whose address is the lower six bits of the instruction word.

2b5e Add constant (adc), add to the contents of the accumulator the contents of the cell following the instruction--and get the next instruction from the cell following that.

2b5f Add forward (adf), add to the contents of the accumulator the contents of the cell that is forward of the instruction cell by the six-bit number found in the lower half of the instruction word.

2b5g Add backward (adb), add to the contents of the

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART C -- ON-LINE COMPUTER EQUIPMENT

accumulator the contents of the cell that is backward from the instruction cell by the six-bit number found in the lower half of the instruction word.

2c Interrupt feature:

2c1 Four independent sources, two internal and two external, may cause an interrupt of what the computer is currently doing.

2c2 Interrupt signal causes contents of accumulator to be put into special cell, and the computer to get its next instruction from the succeeding cell.

2c3 The special cells, for the four sources, are cells 10, 20, 30, and 40--hence the sources are generally called the interrupt-10, interrupt-20, interrupt-30, and interrupt-40 sources.

2c4 Programmer can lock out these interrupt inputs programmatically.

2c5 If interrupts are not locked out, interruption occurs at completion of current instruction.

2d Input-output provision:

2d1 Two input-output channels that can operate independently--termed "normal" and "buffer."

2d2 Normal works as one expects--give a command to input or output and the computer waits until the job is done before it goes on to do further work.

2d3 Buffer works independently of the normal instruction cycles. Give an instruction for a buffer in or out and the main sequence of operations will continue while this input or output is being carried out. Every time the buffer channel needs access to the memory it steals a cycle from the main program sequence without otherwise bothering it. At the end of the buffer operation, an interrupt-20 automatically occurs--and the programmer has had to be ready with the appropriate instructions starting at cell 21 to take care of this.

2d4 After a device has been selected, all subsequent input (or output, if selection was for output) instructions operate with that device.

2d5 There is a family of single-word transfer commands

that send or receive one word per instruction.

2d6 There is family of block-transfer commands that will send or receive an arbitrary-length block to or from consecutive cells of memory, at the rate determined by the external device.

3 PERIPHERAL EQUIPMENT:

3a For any device, transfer to and from the computer (on either channel) can be accomplished by single-word-at-a-time commands, or by block-transfer commands.

3b Paper tape reader. Photo-electric. Can read at asynchronous rate up to maximum of 320 frames/sec. Will accept 6-, 7-, or 8-level tape. Always on normal channel.

3c Paper-tape punch. A Teletype product, punching 8-level oiled tape. Can punch asynchronously up to maximum of about 120 frames/sec. Always on normal channel.

3d On-line typewriter. IBM typewriter, with CDC interface. Can couple to either channel.

3e Character generator.

3e1 Several modes of operation, in which it interprets differently the words sent from the computer.

3e2 The mode is determined by the program code used to select the character generator for coupling to the output channel.

3e3 The mode we use for text interprets the words following the select instruction as follows:

3e3a The first word specifies vertical position (nine bits) and the least-significant three bits of horizontal position.

3e3b All succeeding words (until another select instruction) specify a character to be displayed (with 6 bits) at the vertical location already designated, and the most-significant six bits of the horizontal position.

3e3c This allows a whole line of characters to be outputted as a block following a select instruction which specifies the vertical position of the line.

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART C -- ON-LINE COMPUTER EQUIPMENT

3e4 There is a repertoire of 43 characters to select from.

3e5 Characters are generated in an asynchronous operation that takes a maximum of 6 microseconds--but the output channel cannot deliver words to the output in less than about 17 microsecond intervals--so we have a maximum generation rate limited by this factor of a little less than 60,000 characters per second.

3e6 We display about 1,000 characters maximum on our screen, and run it at a rep rate of 60 frames/sec.

3e7 The new character generator being installed soon will interpret output words as specifying two characters per word, and will double our displayable capacity.

3f Mag tape. A CDC Type 603, compatible with IBM and Burroughs B5500. Programmer can write records of arbitrary length--transport automatically leaves inter-record gaps after stop sending it data. Has end-of-file code that can be put on programatically. Will read forward one record at a time, or back up one record at a time, from a single instruction.

3g Drum--a 32,000-word, fixed-head auxiliary storage device.

3g1 Speed, about 30 rev/sec.

3g2 Can only make access to records--two records per track, 32 tracks.

3g3 Each record holds 512 12-bit words.

3h Special interface and associated devices used by the on-line console.

3h1 Light pen, manufactured by Sanders Associates of Nashua, New Hampshire.

3h1a A photomultiplier tube in the control unit receives light by means of a fiber-optic bundle from a hand-held pen containing a lens which focuses light on the bundle.

3h1b A circle of orange light is projected from the pen to aid in aiming. The source for this light is in the control unit, and light is again transmitted by a

fiber bundle.

3h1c When a light pulse of suitably fast rise time is detected an electrical pulse is generated in the control unit. A switch on the body of the pen unit gates this pulse to the interface logic.

3h1d In the single-pulse mode of operation, only one pulse is produced each time the pen button is depressed and the finder beam goes out to indicate a successful detection.

3h1e In the continuous mode of operation, a pulse is sent to the interface each time a light pulse is detected, as long as the pen button is held down.

3h1f The pulse mode is set by means of a switch on the control box (to which the 34-inch fiber-optic bundle attaches).

3h1g When the interface receives a pulse from the light pen control unit, an interrupt is sent to the computer and the six most-significant bits of the last computer output word are stored. (These six bits represent the horizontal position of the character on the display which produced the light pulse.)

3h2 An analog-to-digital converter, manufactured by Dynamic System Electronics, allows the digitizing, selecting, and inputting to the computer of four different analog input channels. The converter produces nine bits plus sign, with a settling time of 400 microseconds. The converter is used to input positional information from the following operand locating devices:

3h2a A joystick, manufactured by Bowmar Associates, has two potentiometers coupled to a vertical stick. The potentiometers are used as voltage dividers, and produce voltages proportional to the X and Y deflection of the stick from its central location. A switch, actuated by pressing down on the stick, may be used as an input to the computer--to mark operand locations, for example.

3h2b The Grafacon, manufactured by Data Equipment Corporation, consists of a linear potentiometer mounted in a frame which is pivoted on an angular potentiometer. The voltage outputs from the two potentiometers represent polar coordinates about the pivot point. A ball or a pen, mounted on the end of

SECTION III -- ON-LINE TEXT MANIPULATION SYSTEM
PART C -- ON-LINE COMPUTER EQUIPMENT

the linear potentiometer shaft, is moved about by the operator and is depressed to actuate a switch which may be used as a computer input.

3h2c The mouse, made by SRI, consists of two potentiometers mounted in a frame with their shafts orthogonal and a wheel on each shaft. As the frame is moved about a surface the potentiometers resolve the motion into two coordinates. A switch mounted on the frame may be used as a computer input.

3h2d A footpedal, made by SRI, consists of a potentiometer coupled to a pedal which is pivoted at its center. Rocking the foot forward and backward operates the potentiometer; a switch operated by the other foot chooses horizontal or vertical input for the output of this potentiometer.

3h3 The interface provides for input to the computer of external contact closures. The switch circuits are arranged in three groups; a group of 15 are encoded to 4 computer input lines, a group of 7 are encoded to 3 input lines, and a group of 5 are input directly to 5 input lines. Actual input lines are selected by means of a patch-panel to provide flexible assignment of bits in the input word.

3h4 A bell mounted in the on-line console may be rung by a select code from the computer.

3h5 An interrupt for timing purposes may be sent to the computer at a selected rate. A multivibrator in the interface covers an interrupt rate range of approximately 30 to 150 cycles. An external input will accept a rate up to about 5000 cycles.

3h6 All interrupts from the interface may be locked out by a select code from the computer, and enabled by another select code.

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART A -- INTRODUCTION

1 Implementation of the Off-Line System has been funded in part as an in-house project and in part by the Air Force Office of Scientific Research.

1a Development of statement-manipulation techniques and programming on the B5500 were supported by Stanford Research Institute as an Institute Sponsored Research project. Included in this effort was the 160A programming required to translate between tapewriter codes and Burroughs code.

1b Z-Code editing features incorporated into this system were developed and programmed on the 160A under the sponsorship of the Air Force Office of Scientific Research.

2 The Off-Line System was implemented to make available machine-aided text editing and updating on a fast-turn-around basis to a larger community than can be served by the current On-Line System.

3 The Off-Line System makes use of the combined facilities of the CDC 160A computer in the Systems Engineering Laboratory and the Burroughs B5500 computer operated by the Mathematical Sciences Department.

3a Since paper tape provides a convenient medium for entering text, and since the B5500 is not equipped for paper-tape input, the 160A is used to translate paper tape input in Flexowriter or Teletype code to Burroughs code on a magnetic tape.

3b The larger core and drum memories of the B5500 are utilized for rapid access to statements anywhere within a fairly long document to combine text from separate input tapes and/or to restructure the contents of a given document according to commands specified in one or more of the input tapes. Statements are inserted, moved, or replaced essentially by successive modifications of statement-to-statement links defining a path through the document. Separate documents may be spliced end-to-end or merged such that their statements are intermingled. Additional text may be appended to existing statements by means of similar links. The B5500 produces an output magnetic tape in which the document is restructured as specified, with its statements renumbered according to a standard format.

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART A -- INTRODUCTION

3c The 160A converts Burroughs code on the output magnetic tape to Flexowriter code and executes Z-Code editing commands embedded in statements or appended to them during the statement-manipulation process on the B5500. The 160A produces a paper tape that may be listed on the Flexowriter to produce hard copy or entered as input to the On-Line System. The output tape may, of course, also be used as input to a later pass through the Off-Line System for updating or further editing or restructuring.

4 The ability to append Z-Code editing commands (which can reach any point within a statement) during the restructuring process permitted separation of the gross restructuring process from the detailed editing process. Since the latter had been previously programmed on the 160A, this organization minimized the programming effort required to implement the system.

5 Any number of paper tapes may be merged to produce a single document.

6 Any number of documents may be processed in a single batch, up to the capacity of a single magnetic tape (with high-density recording).

7 Statement-manipulating procedures and Z-Code editing functions have been so designed that everything about the eventual output from the Off-Line process can be unambiguously determined by examining the tapewriter input.

7a This principle assures the user that he can edit or otherwise manipulate text material according to the way it appears on the hard-copy listing without risk of error due to non-printing keyboard actions or phantom characters that would throw line, word, or character counts off.

7b Adherence to this principle has made it possible to take "old" documents produced with early versions of the text-editing processes and rework them using later techniques without being trapped by some forgotten (hidden) feature of their machine coding.

FL PROCEDURES

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B1 -- STATEMENT-STRUCTURE MODIFICATION:
PROCEDURES FOR ENTERING MATERIAL

1 Implementation of the Off-Line System has been funded in part as an in-house project and in part by the Air Force Office of Scientific Research.

1a Development of statement-manipulation techniques and programming on the B5500 were supported by Stanford Research Institute as an Institute Sponsored Research project. Included in this effort was the 160A programming required to translate between tapewriter codes and Burroughs code.

1b Z-Code editing features incorporated into this system were developed and programmed on the 160A under the sponsorship of the Air Force Office of Scientific Research.

2 Input is via paper tape prepared on Flexowriter or Teletype machines.

2a "Notes for Orientation of Personnel Preparing Copy for the Off-Line System" is a useful reference for the first-time user.

2b "User Guide to Statement Manipulation in the Off-Line System" is a concise reference for the experienced user.

2c "Z-Code Reference Summary" is a reference document describing editing operations within statements.

2d "Capitalization and Underlining on the Model 33ASR Teletypewriter" is a guide to the use of this machine for the preparation of input material.

3 All tapes should carry the source data in man-readable form, i.e., initials of originator and date in white pencil or gummed label on the tape leader.

3a Tapes to be merged should carry identical source data, i.e., the source data of the original memo.

3b Tapes for different jobs carrying the same initials and date must be identified by serial numbers following the date, i.e., ART 15 FEB 65-1 and ART 15 FEB 65-2.

4 All tapes should be labelled as to the machine code: FLX if prepared on Flexowriter, TTY if prepared on Teletype machine, FL if output from a previous pass through the off-line (FL) system, and NL if output from the on-line system.

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B1 -- STATEMENT-STRUCTURE MODIFICATION:
PROCEDURES FOR ENTERING MATERIAL

- 4a Output from the off-line and on-line systems will normally be in FLX code.
- 5 A single original tape or the primary tape to which others are to be merged need not carry additional information.
- 6 Two types of merge operation are available. In labelling tapes for processing, "merge," in the narrow sense, will be used to refer to tapes carrying data to be interleaved with a primary tape. "Follow" will refer to tapes carrying statement lists to be tacked onto the end of a primary tape. The latter mode permits separate memos repeating some of the same statement numbers to be spliced in sequence to form a longer memo.
- 6a Tapes to be merged with a primary tape should carry the word "merge" and a number indicating the order of merging; thus "merge #1" would be merged with the primary tape before the tape labelled "merge #2."
- 6b Tapes to follow a primary tape should carry the word "follow" and a number (which must be a multiple of 10) to be prefixed to each statement number of the following memo. This number must be larger than the highest principal-statement or heading number of the memo it follows, and it must be distinct from the prefix used for any other "follow" tape to be combined with the same memo. Operation of the prefix is that of placing 10. in front of each statement number in the following memo, if 10 is the prefix designated.
- 7 A brief form on a 3-by-5 card, available at the collection point, must be filled out for each job. This form is self-explanatory.
- 8 The tapes for each job should be stacked on top of the 3-by-5 card at the collection point.
- 9 Normal hard-copy outputs are (1) a B5500 listing, with the Z-Code commands not yet executed, and (2) a Flexowriter listing of the output paper tape, produced after Z-Code execution. The paper tape in FLX code is the machine-readable output.
- 9a At times, the Flexowriter may be a bottleneck in the system. At such times, faster turn-around may be achieved by working with the B5500 listing and not waiting for the Flexowriter listing. Care must be exercised, however, since the Z-Code processing will result in reformatting within statements, so that format on the B5500 listing may not be the same as that on the paper tape.

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B1 -- STATEMENT-STRUCTURE MODIFICATION:
PROCEDURES FOR ENTERING MATERIAL

9b If Z-Code commands are used only to modify immediately adjacent text, i.e., text within the entered statement in which they occur, Z-Code processing can be performed prior to B5500 processing, and the B5500 listing will be "clean." This will not work, of course, for Z-Code commands in APPEND statements that reach into text entered in a previous statement or on another tape.

10 Until the format of the 3-by-5 cards is modified to include a specific place for this information, please write on the card either "Z-Code FIRST" or "Z-Code LAST."

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B2 -- STATEMENT-STRUCTURE MODIFICATION:
PREPARING COPY

1 Implementation of the Off-Line System has been funded in part as an in-house project and in part by the Air Force Office of Scientific Research.

1a Development of statement-manipulation techniques and programming on the B5500 were supported by Stanford Research Institute as an Institute Sponsored Research project. Included in this effort was the 160A programming required to translate between tapewriter codes and Burroughs code.

1b Z-Code editing features incorporated into this system were developed and programmed on the 160A under the sponsorship of the Air Force Office of Scientific Research.

2 A STATEMENT is a segment of text headed by a statement number preceded by two carriage returns (or, on the Teletype, two line feeds).

2a All elements of the text, including the Source, Title, Abstract, etc., must be in statement format; that is, they must be preceded by two carriage returns (or two line feeds) and appropriate statement numbers.

2b The first characters entered on any tape must be preceded by two carriage returns (or line feeds).

3 A STATEMENT NUMBER is an alternating sequence of numbers (one or more digits) and letters (doubled, tripled, etc. if necessary).

3a The first symbol of a statement number must be a numerical digit.

3b Literal elements of statement numbers, a, b, c, etc., must be lower case. Slashes (/) and plus signs (+) within or preceding a statement number will invalidate the number.

3c Statements headed by numbers alone designate the highest level in the text structure, either the major headings or the principal lead statements.

3d Statement numbers of the form 2a, 2b, 2c, etc. designate elements of a statement list, or substructure, subordinate to the head statement designated by the number 2 alone.

3d1 If the number of items in a statement list carrying a letter as its last character exceeds 26, letters are doubled up according to the following convention: 2x,

FL COPY PREP

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B2 -- STATEMENT-STRUCTURE MODIFICATION:
PREPARING COPY

2y, 2z, 2aa, 2ab, 2ac, . . . 2az, 2ba, etc.

3e Statement numbers of the form 2b1, 2b2, 2b3, etc. designate elements of a statement list, or substructure, subordinate to the head statement designated by the statement number 2b.

3e1 Numerical sequences may be as long as required:
2b8, 2b9, 2b10, 2b11, . . . 2b99, 2b100, etc.

3f Regardless of their order in the input text, the B5500 will output statements in the order determined by their statement numbers.

3f1 In the reordering of statements according to statement number, the substructure under each statement will be outputted directly following that statement, and this rule will govern down to the lowest level of the structure, as in this document.

4 Statements may be interpolated into an existing list by utilizing the following conventions:

4a A statement to be inserted between major headings 2 and 3 and of equal rank with them may be assigned the statement number 2.5 (the 5 could be any digit or decimal number). The B5500 will renumber this inserted statement 3, change the former 3 to 4, etc. all the way to the end of the list. Furthermore, it will make the same changes to the first numbers of all subordinate statements, so that each heading statement will retain its own substructure.

4a1 If several statements are to be interpolated between two existing statements, they may be numbered 2.3, 2.4, 2.5, 2.52, 2.6, etc., and they will be inserted in order of their decimal values; that is, 2.52 would come after 2.5 and before 2.6 in the B5500 output, regardless of their order in the input text.

4a2 If an inserted statement should carry a substructure of subordinate statements, they may be designated as follows: 2.52a, 2.52b, 2.52c, etc. When the 2.52 is changed to a whole integer in renumbering, the subordinate statement numbers will be altered to agree, so that the substructure will follow the referenced statement.

4b A statement to be inserted between 2b and 2c and of equal rank with them may be assigned statement number 2b.m (the m could be any letter of the alphabet or string of

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B2 -- STATEMENT-STRUCTURE MODIFICATION:
PREPARING COPY

letters.)). The B5500 will renumber this inserted statement 2c, change the former 2c to 2d, etc. all the way to the end of the substructure list under heading 2. Furthermore, it will make the same changes to the corresponding letters in the numbers of all subordinate statements involved, so that each statement will retain its own substructure.

4b1 If several statements are to be interpolated between two existing statements with final literals in their statement numbers, they may be designated as follows: 2b.a, 2b.c, 2b.m, 2b.mb, 2b.n, etc., and they will be inserted in alphabetical order, treating second letters as interpolations between first-letter designations; that is, 2b.mb would come after 2b.m and before 2b.n in the B5500 output, regardless of their order in the input text. (This amounts to a decimal interpretation of the literal string, consistent with the interpretation of the numerical string.)

4b2 If an inserted statement that will carry a final literal in its statement number should carry a substructure of subordinate statements, they may be designated as follows: 2b.mbl, 2b.mb2, 2b.mb3, etc. When the 2b.mb is changed to a number followed by a simple literal in renumbering, the subordinate statement numbers will be altered to agree, so that the substructure will follow the referenced statement.

4c The conventions described above may be utilized at all levels of the text structure. If the level in which interpolation is to take place is designated by statement numbers with final numerical symbols, the interpolation string is numerical. If the level in which interpolation is to take place is designated by statement numbers with final alphabetical symbols, the interpolation string is alphabetical.

4d In cases where it becomes necessary to insert a statement before the first item of a list or sublist, the following convention is useful: 1,5 will be renumbered 1, with all subsequent numbers increased, so that the list is pushed down. 2a,m will be renumbered 2a, with all subsequent second literals in the list advanced one letter, thus pushing down this sublist. All other conventions discussed in 3a thru 3c hold when the period (.) is replaced by the comma (,). Interpolation now takes place before the statement whose number precedes the comma, rather than after the statement whose number precedes the period. The relative order of multiple insertions is governed by the same decimal interpretation as when the period is used; i.e., the comma does not reverse the sense of the

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B2 -- STATEMENT-STRUCTURE MODIFICATION:
PREPARING COPY

interpolation, it merely designates interpolation into the preceding rather than the following interval.

4e If two statements should be inadvertently entered with the same number, the statement entered last will follow the first, and they will be renumbered consecutively.

5 DELETE, REPLACE, MOVE, and APPEND Operations are achieved by utilizing statements with coded instructions tacked onto their statement numbers.

5a Command codes are literal elements, d, dt, dl, r, m, and a, following a colon (:). These literal elements must be lower case. Slashes (/) and plus signs (+) within the command structure will invalidate the command.

5b Each of the following commands must be entered as a separate statement; that is, the coded statement number must follow a double carriage return (or double line feed).

5c The coded statement number 2b1:d will delete statement 2b1 wherever it exists, either in the original copy or in the correction copy.

5c1 Deletion of a statement automatically deletes all of the substructure under that statement; thus the command 2b1:d will delete not only statement 2b1 but all statements with 2b1 followed by any combination of letters and numbers. It will remove 2b1a, 2b1a1, 2b1b, etc.

5c2 When a statement, with its substructure, is deleted, the B5500 will renumber the remaining elements of the list and the substructure statements under them, so that there will be no discontinuity in the number designations.

5c3 The delete code may be used to delete a statement that is itself a delete command. For instance, if the delete command 2b1:d has been entered anywhere in text as a statement, the statement 2b1:d:d will remove the delete command, and the original statement 2b1 will stand.

5d The coded statement number 2b1;la:dt will delete statement 2b1 and any and all statements following it in the input text up to and including statement la. (In this example, it is assumed that text is being entered out of order and that there is a statement 2b1, followed later on by a statement la, with any number of intervening statements.) This :dt code is used to remove statements from

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B2 -- STATEMENT-STRUCTURE MODIFICATION:
PREPARING COPY

the text material on the tape currently being prepared on the tapewriter, whether new material or correction copy. The :dt code will not reach material on any previously processed tape with which the currently prepared text is to be merged. Neither will it reach beyond the segment of input text bounded by the referenced statements. A statement numbered 2b1a, for instance, would be deleted along with its heading statement 2b1 only if statement 2b1a lay between 2b1 and 1a in the input text; otherwise it would remain.

5e The coded statement number 2b1;2d:d1 (final character is letter "l") will delete statement 2b1 and any and all statements following it in structured order, up to and including statement 2d and all of the substructure under 2d. The statements between 2b1 and 2d, and the substructure of 2d, may have been entered on separate tapes, intermixed with any other statements, etc. Wherever they exist in the structured or unstructured text, items headed by statement numbers beginning with 2b1, 2b2, 2b3, . . . 2c, and 2d will be deleted. Statements with numbers out of this range will not be deleted, even though they may be intermixed in the text.

5e1 When a group of statements, with their substructure, is deleted, the B5500 will renumber the remaining elements of the list and the substructure statements under them, so that there will be no discontinuity in the number designations.

5f The coded statement number 2b1:m 2b3.5 will renumber the statement numbered 2b1 with the number 2b3.5 and thus cause it to be moved to a position in the structure between the statements previously numbered 2b3 and 2b4. Since the original 2b1 is now removed, however, all of these numbers may be changed. The single space following the code letter "m" is required. The second referenced statement number, 2b3.5, need not be an interpolation number; it could be 2b6, 2c, 3f, or any other.

5g The coded statement number and following literal string 2b1:r Now is the time for all good men . . will replace the previous text of the statement numbered 2b1 with the text "Now is the time for all good men . ." The replacement code :r will not affect any other statement except the referenced one. Replacement is complete, and cannot be partial; that is, the whole of the statement is removed and replaced by the literal string following the coded statement number.

5h The coded statement number and following literal string

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B2 -- STATEMENT-STRUCTURE MODIFICATION:
PREPARING COPY

2b1:a Now is the time for all good men . . will append the words "Now is the time for all good men . ." to the end of statement 2b1. All of the former statement remains intact, and the addition will be made only at the end. A single space following the code letter "a" is required. Any additional spaces preceding the literal string will appear as a spacing gap between the end of the former text and the beginning of the addition. If one desires to leave two spaces before an added sentence, the first letter of the sentence should be separated from the code letter "a" by three spaces.

5h1 Since, in the current system, Z-Code processing will follow statement processing on the B5500, and since the range of Z-Code commands will be limited to one statement, the :a operation may be used to append Z-Code commands to statements, providing for deletion and insertion of text within selected statements. Note that the Z-Code INSERTION command must be followed by a spacing character, and that this spacing character will be deleted when the command is executed. In order to avoid deletion of one of the required carriage returns at the end of the statement, one should follow the insertion command by one or more spaces. In the event that a carriage return is inadvertently entered immediately following an insertion command, follow it with at least two more carriage returns.

6 Since Z-Code processing within statements will follow statement processing on the B5500, Z-Code commands cannot be used to delete, modify, or insert statement numbers or coded statement numbers constituting commands to the statement-manipulating system. Therefore, the following conventions have been implemented to permit modification of statement numbers (either uncoded, or coded with command symbols):

6a If an error is recognized while typing a statement number, and only the last few symbols are in error, the PERCENT (%) sign may be typed. Each %-sign will delete one character backward in the statement number. Thus lb2a%b will be corrected to read lb2b, and lb2a%%3a will be corrected to read lb3a. Command symbols may be similarly corrected; for instance, lb2;lc:dl%t will be corrected to read lb2;lc:dt.

6b If an error is recognized while typing a statement number and it would be just as well to start over from scratch, the DOLLAR (\$) sign may be typed. The \$-sign deletes all that has been typed of the statement number (and

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B2 -- STATEMENT-STRUCTURE MODIFICATION:
PREPARING COPY

command code), back to the double carriage return that preceded it. Thus lb2a\$lb3a will be corrected to read lb3a, and lb2;le:d1\$lc2;le:dt will be corrected to read lc2;le:dt.

6c The %-sign and \$-sign delete commands must be made within the statement number or command, and thus depend on catching the error before going past it by too many symbols. If the incorrect statement number or command has been completed and a spacing character typed, it may be corrected or deleted by a later command constituting a separate statement.

6c1 A MOVE command may be used to correct a statement number. For instance, if a statement has been entered with an incorrect number, lb3, a later statement consisting of the move command lb3:m lc3 will have the effect of correcting the statement number to read lc3.

6c2 An incorrect command code or an incorrect statement number in a command can best be corrected by deletion and re-entry. Deletion is accomplished by repeating the incorrect command followed by the symbols :d as a separate statement. In such cases, the correct command will then have to be typed as another separate statement.

FL USER'S GUIDE

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B3 -- STATEMENT-STRUCTURE MODIFICATION:
USER GUIDE

1 Implementation of the Off-Line System has been funded in part as an in-house project and in part by the Air Force Office of Scientific Research.

1a Development of statement-manipulation techniques and programming on the B5500 were supported by Stanford Research Institute as an Institute Sponsored Research project. Included in this effort was the 160A programming required to translate between tapewriter codes and Burroughs code.

1b Z-Code editing features incorporated into this system were developed and programmed on the 160A under the sponsorship of the Air Force Office of Scientific Research.

2 Processing conventions:

2a The desired operation is completely specified by the first word of each entry statement--generally the standard location number or some variant on this.

2b When a statement is deleted, its substructure is deleted.

2c When a statement is moved, its substructure is moved with it.

3 User processes:

3a Insertion:

3a1 If, for a given statement, its location number has only alphanumerics, periods, or commas in it, and is followed normally, i.e., by a spacing gap, then that statement is to be inserted as a new statement in the location implied by the location number.

3a1a If several statements are thus assigned the same location numbers, their ordering in the eventual structure will be the order of their entry. They will be given consecutive location numbers in the final renumbering.

3a2 Allow use of interpolative numbering in location numbers to designate eventual location of statements

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B3 -- STATEMENT-STRUCTURE MODIFICATION:
USER GUIDE

being referenced.

3a2a Let 2a4.5 (or 4a.d) designate a location number coming after 2a4 (or 4a) in eventual interpolative order.

3a2b Let 2a1,2 (or 4a,g) designate a location number that comes before 2a1 (or 4a) in eventual interpolation order.

3a2b1 Interpret the characters after the comma as though they were positive-ordered designators that started from the predecessor location number (even though there may be no predecessor statement--i.e., the statement referenced is a head statement).

3a2b2 Assume that 2a1,3 (or 3a,c) would come before 2a1,5 (or 3a,e).

3a2c Compound interpolation is allowed: e.g., 2a4.5.2 (4a.d.b) designates a location number which would be between 2a4.5 (4a.d) and 2a4.6 (4a.e).

3a2d Multiple-character fields are not to be confused with interpolation designation: e.g., 2a3.12 and 2a3.25 represent the twelfth and twenty-fifth interpolative positions between 2a3 and 2a4--and are not the second and fifth positions between 2a3.1 and 2a3.2, or 2a3.2 and 2a3.3.

3b Appending to and modifying a prior statement:

3b1 Let a statement beginning with LN1:A designate that the rest of this statement will be appended immediately after the last printing character of STM LN1.

3b1a A SPACINGAP must appear after the "A" in the append command.

3b1b The processor removes this SPACINGAP during the append operation (before the Z-code processes are executed).

3b2 Any Z-Codes included in the appended string will be executed, treating the new composite statement as a whole, after all of the inserting, appending, deleting, and moving of statements has been done.

3c Replacement:

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B3 -- STATEMENT-STRUCTURE MODIFICATION;
USER GUIDE

3c1 Let LN1:R designate that the entire text of STM LN1 is to be replaced by the text following the R.

3c2 The new STM LN1 will have the same location number (LN1), with new text.

3d Deletion:

3d1 Let LN1:D designate that STM LN1 be deleted.

3d2 Let LN1;LN2:DT designate that the input text string including and between STM LN1 and STM LN2 is to be deleted.

3d2a This deletes all statements, of any kind and level, in this string.

3d3 A delete command can operate upon a prior delete-command statement by using as the reference location number the entire compound word heading that statement.

3d3a For example, STM 2a4b is deleted by a statement headed 2a4b:d. But this delete command can itself be deleted by a statement headed 2a4b:d:d.

3e Moving statements and structure sections:

3e1 Let LN2:M LN1 designate that, to the structure location specified by LN1, the statement STM LN2 and its entire substructure is to be moved.

3e1a The statement STM LN2, its substructure, and the lists and substructures displaced by this move, will all be renumbered after the deleting, inserting, and moving operations are done.

3f Correcting statement-manipulation commands:

3f1 Let \$ in a location number (in the op-code part of our statement-manipulation) designate that the \$ and all characters up to it, are to be deleted by the B5500 processor before the command is interpreted.

3f2 Let % in the location number designate that both the % and the character just preceding it are to be deleted by the B5500 processor before the command is interpreted.

3f3 Before interpreting any command statement, the

processor will begin at the left end of the location number and proceed to the right, character by character, looking for \$ or % characters, and executing them immediately.

3f3a This means that n successive % characters will delete the n preceding characters.

3g General considerations:

3g1 The new numbers, appearing on the subsequent printout, will have no interpolation numbers.

3g2 The user may consider that the actual moving is not done until the very last of the processing for the whole job. Thus, for instance, after a LN1:M LN2 command, he can refer to STM ln1 or any statements of its substructure by their location numbers as seen in the "original" hard copy.

3g3 It may help if the user thinks of these commands as establishing new structural linkages (i.e., to list-successor and sublist-head statements) between existing statements, with renumbering to be done after all such new linkages are established.

3g4 The compound location numbers that effect relocation and deletion of other statements are to be the heads of empty statements.

3g5 It is useful to remember that the processor makes two passes through the entire input text.

3g5a First pass, backwards, executing only delete commands.

3g5b Second pass, forward, executing all other commands.

3h Things to be careful about:

3h1 Use no Z-codes in the location number (or command)--the % and \$ signs are the only acceptable ways to make corrections in the location number.

3h2 For statements that are given the same location number (not a forbidden event--they will be inserted in order and given new numbers), the processor will hang up if one tries to refer to that location number for a move, delete, append, or replace.

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B3 -- STATEMENT-STRUCTURE MODIFICATION:
USER GUIDE

3h3 Tabs appearing at the beginning of the line (i.e., immediately after a carriage return) will be removed.

4 Special Features:

4a Merging of two records:

4a1 Assume that the location numbers of the two records are independent of one another and that for each record they began with 1.

4a2 One may designate to the operator to load the second tape with a prefix integer, N.

4a3 Upon loading the second tape, the operator keys this integer in as a special parameter, and all statements in that record will have a prefix attached to the front of their location numbers composed of the integer N followed by a PERIOD.

4a4 The user would then write a new third tape to specify the manner in which the contents of the second tape are to be integrated with those of the first tape.

4a4a When referencing statements of the second record, the user must be careful to designate their location numbers with the appropriate prefix which he specified.

4b Multiple sequence input entry:

4b1 A user sitting at his own tape-punching typewriter preparing material dealing with a number of independent records, often finds that new thoughts occur for the modification of one record while he is typing on the modification for another.

4b2 The feature here described allows him in such a situation to interrupt the sequence being composed for the one record and introduce, on the same paper tape input, new statements for the sequence referring to the other record.

4b3 To use this feature, one designates an integer job number for each of these independent input sequences which he wishes to use. (He will communicate to the operator which paper tape records each of these corresponds to.)

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B3 -- STATEMENT-STRUCTURE MODIFICATION:
USER GUIDE

4b4 When typing his input, the user may insert at any point a statement beginning with a # character followed immediately by an integer and then a SPACINGAP.

4b4a The integer designates to which record the following statements are to refer.

4b4b In this sequence-break statement, any comment-type text may follow the SPACINGAP, and will be ignored by the processor.

4b5 The operator will insert the necessary parameters at load time so that for each of the independent input records, the processor will scan the input tape and extract the statements referring to that record.

FL OPER

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B4 -- STATEMENT-STRUCTURE MODIFICATION:
OPERATOR INSTRUCTIONS

1 Implementation of the Off-Line System has been funded in part as an in-house project and in part by the Air Force Office of Scientific Research.

1a Development of statement-manipulation techniques and programming on the B5500 were supported by Stanford Research Institute as an Institute Sponsored Research project. Included in this effort was the 160A programming required to translate between tapewriter codes and Burroughs code.

1b Z-Code editing features incorporated into this system were developed and programmed on the 160A under the Air Force Office of Scientific Research.

2 In the current version of the off-line system the following steps are taken to process information:

2a Tapes (either Teletype or Flexowriter) are first converted to magnetic tape using the CONVERT program for the 160a.

2b Statement manipulation commands are then executed on the B5500.

2c The MAG-TAPE ZCODE program is used to execute Z-code commands on the statements and output a Flexowriter paper tape, which can be listed and recycled through the system.

3 Use of the 160a for converting paper tapes to magnetic tapes:

3a Turn on power.

3b Master clear.

3c Turn on paper tape reader.

3d Put a magnetic tape with a write ring on the tape unit; put magnetic tape unit on 0; and connect tape to normal channel.

3e (LOAD) Load CONVERT program paper tape at 0000.

3f Master clear.

3g (RUN) Put data tape in reader and run.

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B4 -- STATEMENT-STRUCTURE MODIFICATION:
OPERATOR INSTRUCTIONS

3h STOPS: The computer will halt at one of the following locations:

3h1 0727: System has failed to clear magnetic tape parity error. Do not master clear. Reset run switch.

3h2 0254 or 0754: The data tape has been processed. Do not master clear. Clear all jump switches. Do one of the following:

3h2a If the next tape is to be "merged" (same job), put 0000 in A and go to(RUN).

3h2b If the next tape is to "follow" (same job), enter prefix in A and go to(RUN).

3h2c If the next tape is not to be merged (different job), put 0001 in A and go to(RUN).

3h2d If there are no more tapes to be processed, put 0002 in A and run.

3h3 0153: Tape did not start with a carriage return. If the paper tape is a Teletype tape, reset run switch. Do not master clear. If the paper tape is a Flex tape, set selective jump switch 2 and reset run switch. Do not master clear.

3h4 0321: Normal completion of processing.

3h5 Any other stops: Computer error. Go to(LOAD) and start processing over.

4 Use of the B5500 for statement manipulation:

4a Take write ring off tape (input tape of B5500).

4b Put write ring on another tape (to be output tape of the B5500).

4c Carry both magnetic tapes to the computation center.

4d Fill out an operator card as follows:

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART B4 -- STATEMENT-STRUCTURE MODIFICATION:
OPERATOR INSTRUCTIONS

COMPUTER REQUEST CARD

PHONE	NAME <i>J. Lass</i>	EST. TIME	MAX. TIME	CDS. OUT	PAPER
JOB NUMBER <i>H847</i>	INSTRUCTIONS <i>Purge SEL 8. Return back tapes & listing</i>				
TIME ON					
TIME OFF					

REEL NO. ▶	SEL 6	SEL 8							
READ	READ	READ	READ	READ	READ	READ	READ	READ	READ
WRITE	WRITE	WRITE	WRITE	WRITE	WRITE	WRITE	WRITE	WRITE	WRITE
DENSITY	LO	LO	LO	LO	LO	LO	LO	LO	LO
LO HI		(HI)	(HI)	(HI)	HI	HI	HI	HI	HI
TAPE ID ▶	<i>Z-code Z-out Search</i>								
OK TO PURGE ▶		X	X						

SRI-2712 8

4e Put card deck, both tapes, and instruction card on the table for B5500 input jobs.

4f Pick up both tapes, output, and card deck after processing is complete on the B5500. The processed data is on the magnetic tape with the write ring.

5 Use of the 160a for final processing:

5a Turn on power.

5b Master clear.

5c Turn on paper tape punch.

5d Turn on magnetic tape unit 1, load data magnetic tape in the magnetic tape unit, and connect tape on buffer channel.

5e Load MAG-TAPE ZCODE program paper tape at 0000.

5f (NEXT) Master clear.

5g Put run switch in run position.

5h Normal stop. Computer will come to a normal stop after the entire data on magnetic tape has been processed and the computer has punched a Flexowriter paper tape. If another magnetic tape is to be processed, go to(NEXT).

5i If computer stops before processing is complete, reset run switch. Do not master clear.

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART C1 -- EDITING WITHIN STATEMENTS:
Z-CODE REFERENCE SUMMARY

1 The Z-Code editing features of the off-line system were developed and programmed for the 160A under the sponsorship of the Air Force Office of Scientific Research.

2 DEFINITIONS

2a Printing Character = any symbol that prints out on the tapewriter (alphanumeric, punctuation, mathematical symbol, etc.).

2b Non-Printing Character = any command function that records on tape but does not print out on the tapewriter (space, carriage return, tab, backspace, etc.).

2c Word = printing character or unbroken string of printing characters isolated by non-printing characters.

2d Gap = non-printing character or unbroken string of non-printing characters bounded by printing characters.

2e Line = character string initiated by a carriage return. A line may be empty (two carriage returns in sequence).

2f Statement = segment of text within reach of Z-Code editing commands. The statement delimiter consists of two carriage returns (or line feeds on the Teletype) and a statement number.

3 CONVENTIONS USED IN THIS WRITEUP

3a All printing characters used in Z-Code descriptions stand for themselves except the letter N and N followed by an integer.

3b The letter N will denote a general integer whose value will specify a number of lines, words, characters, or tab stops in a Z-Code control string.

3c The letter N followed by an integer, N1, N2, N3, etc., will denote a subscripted N, that is, a general integer. Subscripted N will be used in expressions or discussions involving two or more integers that can take on independent values.

4 STRUCTURE OF A Z-CODE EDITING COMMAND

4a The computer recognizes a Z-Code editing command by the occurrence of a letter Z followed by an integer.

FL ZCODE

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART C1 -- EDITING WITHIN STATEMENTS:
Z-CODE REFERENCE SUMMARY

4b Initiation and termination of specific Z-Code commands is either explicit, involving specified symbols, or implicit (e.g., terminated by completion of a control string).

4c Point of editing within the current statement is designated by a control string of general form N1LN2WN3C specifying a count of N1 lines (L), N2 words (W), and N3 characters (C).

4d An editing command may contain a data string (text and/or other characters) delimited by parentheses (xxx...xxx).

5 EXECUTION OF Z-CODE EDITING COMMANDS

5a The computer searches backward through a statement, finding and executing Z-Code commands on a "last entered, first executed" basis.

5b Z-Code commands are treated as normal text words when they occur at the point of editing; hence, later commands can delete or modify earlier commands.

5c After execution of all Z-Code commands in a statement, text is "closed up" by replacing line-initiation commands with spaces or spaces with line-initiation commands as required to justify text to left margin and fill out complete lines.

6 DELETION: ZNL, ZNW, ZNC, ZN1LN2WN3C

6a ZNL deletes N lines backward in text, counting as the first line the one in which the Z-Code command occurs.

6a1 Line deletion is executed by deleting backward in text until N carriage returns have been removed.

6a2 Deletion of a carriage return automatically removes any tabs and/or spaces preceding the carriage return.

6a3 Point of reentry after line deletion is immediately following the last printing character on the preceding line. If that line is empty, the reentry point will follow the carriage return.

6b ZNW deletes N words backward in text, counting as the first word the Z-Code command or any unbroken string of printing characters of which it is a part.

6b1 Word deletion is executed by deleting backward in

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM

PART C1 -- EDITING WITHIN STATEMENTS:

Z-CODE REFERENCE SUMMARY

text until N gaps have been removed.

6b2 Deletion of a word thus removes the gap preceding that word. The deleted gap may contain any number of carriage returns.

6b3 Point of reentry after word deletion is immediately following the last printing character of the preceding word.

6c ZNC deletes the Z-Code command and N characters immediately preceding the Z-Code.

6c1 Both printing and non-printing characters are counted, including spaces and carriage returns. Each space introduced by a tab is counted as a separate character.

6c2 Deletion of a carriage return automatically removes any tabs and/or spaces preceding the carriage return; thus after deletion of a carriage return, the next character counted will be the last printing character on the preceding line. If that line is empty, the next character counted will be its carriage return.

6c3 Point of reentry after character deletion is immediately following the last surviving character, which may be either a printing or non-printing character.

6d ZN1LN2WN3C deletes N1 lines, N2 words, and N3 characters backward in text according to the conventions described above for the separate commands.

6d1 Order of execution is line deletion, followed by word deletion, followed by character deletion, regardless of order within the control string of the command.

6d2 Line counting begins with the line that includes the Z-Code command. Word counting begins with the last word of the last surviving line. Character counting begins with the last printing character of the last surviving word.

6d3 Point of reentry after compound deletion is immediately following the last surviving character.

6e The deletion command is implicitly terminated. The next character immediately following the control string will appear at the point of reentry following execution of the deletion command. This may be either a printing or

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART C1 -- EDITING WITHIN STATEMENTS:
Z-CODE REFERENCE SUMMARY

non-printing character. Only characters of the form NL, NW, or NC must be excluded, since they would be interpreted as additions or amendments to the control string.

6f Since text is scanned backward toward the beginning during execution of deletion commands without interpreting deleted words, earlier editing commands may be deleted before execution, and thus will never be executed.

7 INSERTION: Z.IN1LN2WN3C(XXX...XXX)Z.2I gap

7a *c The periods (.) inserted in the above example and in similar examples to follow are to be ignored. Their sole purpose is to "spoil" the Z-Code command so that the example will remain in text and not be interpreted as a valid editing command, since this memo is being prepared using Z-Code editing techniques.

7b Insertion commands are explicitly initiated by the character string Z.1I followed immediately by a control string.

7c The control string N1LN2WN3C is of the same form as that of a deletion command, but its interpretation is different:

7c1 Non-zero line, word, and/or character counts in the control string key on the beginning of the statement, line, and/or word, and counting proceeds forward in text during execution.

7c1a If none of the integers in the control string N1LN2WN3C is zero, the point of insertion will be immediately following the N3-th character of the N2-th word of the N1-th line of the current statement.

7c1b A zero character count (0C) in the above control string would place the point of insertion before the first character of the N2-th word of the N1-th line of the statement, i.e., following the gap that precedes the N2-th word of that line.

7c2 Omitted line, word, and/or character counts in the control string designate the last line of a statement, last word of a line, and/or last character of a word.

7c2a Omitted or zero line count specifies insertion within the line containing the Z-Code command.

7c2b Omitted or zero word count specifies insertion

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART C1 -- EDITING WITHIN STATEMENTS:
Z-CODE REFERENCE SUMMARY

within or adjacent to the last word of the designated line.

7c2c Omitted character count specifies insertion immediately following the last character of a designated word.

7c2d Note that omitted character count and zero character count (OC) are distinct and produce different results.

7d An insertion string, enclosed in parentheses (xxx...xxx), follows the control string.

7d1 The insertion may be any string of characters, including text, punctuation, control characters, and Z-Code deletion commands, but excluding Z-Code insertion commands.

7d2 A deletion command within an insertion string will be inserted at the specified point in text, to be executed later when the translator has scanned backward to that point.

7d3 If spaces are required to separate an insertion from adjacent text, they must be included in the parentheses.

7e Insertion commands are explicitly terminated by the character string Z.2I, followed immediately by a gap or the control string of an additional insertion command.

7e1 Multiple insertion commands are formed by following the character string Z.2I by the control string, insertion string, and terminating string of each successive command, without repeating the Z.1I initiating string.

7e2 A gap must follow the final Z.2I of a multiple insertion command.

7e3 The first non-printing character following an insertion command will be deleted with the command statement when it is executed; hence, this should be a space or extra carriage return not required in the ultimate formatting during close-up of text.

7f Restrictions on the formation of insertion commands:

7f1 Deletion commands embedded in the insertion command,

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART C1 -- EDITING WITHIN STATEMENTS:
Z-CODE REFERENCE SUMMARY

other than within the insertion string, will invalidate the insertion command.

7f2 Carriage returns embedded in the insertion command, other than within the insertion string, will invalidate the insertion command if they follow non-alphabetic characters but will be ignored if they follow alphabetic characters.

8 CONTROL STRING DETAILS COMMON TO DELETION AND INSERTION:

8a Any unbroken string consisting solely of integers alternating with any of the letters L, W, and C that begins with an integer and ends with one of the letters is a semantically valid control string.

8b Line, word, and character counts specified by a control string are the integers just preceding the last occurrence of the letters L, W, and C, respectively. Prior entries of a repeated specification are ignored.

8c Order of occurrence of L, W, and C in a control string may be completely arbitrary. Execution will be the same, regardless of the order in which the final specifications are made.

8d Amendment of specifications during construction of a control string may thus be achieved by merely appending revised specifications to the end of the string.

9 TABULATION

9a Tab stops are "set" in the software package as being at every eighth character position from the left margin.

9b Occurrence of a tab character will insert spaces as required so that the following character will occupy the character position designated by the next tab stop.

9c LEFT margin control: ZNT

9c1 The Z-Code command ZNT, where N is an integer, establishes a "normal" left-hand margin at the N-th tab stop. The command itself will be removed from text in the editing process. The "normal" left margin established by this command controls formatting of all following text until this formatting specification is revised or removed by another ZNT command, where N is another integer or zero.

SECTION IV -- OFF-LINE TEXT MANIPULATION SYSTEM
PART C1 -- EDITING WITHIN STATEMENTS:
Z-CODE REFERENCE SUMMARY

9c2 The effect of this formatting command is to establish a "normal" line-initiation string consisting of a carriage return followed by N tab characters.

9c3 Carriage returns embedded in running text must be followed by N tab operations if ZNT has been specified and it is intended that edited text be justified to this "normal" left margin.

9c4 When text is "closed up" following execution of all Z-Code commands, "normal" line-initiation strings may be replaced by spaces and spaces by "normal" line-initiation strings as required to justify text to the "normal" left margin and fill out complete lines.

9c5 Line-initiation strings consisting of carriage returns alone or carriage returns followed by other than N tab characters will not be deleted or altered in the "close-up" process.

FL TY